

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Програмної інженерії  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

рівень вищої освіти другий (магістерський)

### Дослідження генетичних алгоритмів та їх застосування у сфері торгівлі

(тема)

Виконав:

Студент 2 курсу, групи ІПЗМ-19-2  
Чолак Костянтин Юрійович

(прізвище, ініціали)

Спеціальність 121, «Інженерія програмного  
забезпечення»

(код і повна назва спеціальності)

Тип програми Освітньо – наукова

(освітньо-професійна або освітньо-наукова)

Керівник проф. Власенко Л. А.

(посада, прізвище)

Допускається до захисту

Зав. кафедри

(підпис)

З.В. Дудар

(прізвище, ініціали)

2021 р.

## Харківський національний університет радіоелектроніки

Факультет Центр післядипломної освіти  
(повна назва)

Кафедра Програмної інженерії  
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121, «Інженерія програмного забезпечення»  
(код і повна назва спеціальності)

Тип програми Освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2021р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Чолака Костянтина Юрійовича  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження генетичних алгоритмів та їх застосування у сфері торгівлі

затверджена наказом університету від 26.03.2021 № 386 Ст

2. Термін подання роботи до екзаменаційної комісії 14 травня 2021р.

3. Вихідні дані до роботи методи визначення ефективності налаштування торгових стратегій

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз сучасного стану розвитку у проблемній галузі, що розглядається, постановка задач дослідження.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій вступ, актуальність теми дослідження, мета дослідження, постановка задачі, найпоширеніші торгові технічні індикатори, метрики

налаштування торгових стратегій, результати досліджень, рекомендації, висновки.

#### 6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Власенко Л. А.		13.05.21

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	25.01.2021-25.02.2021	виконано
2	Постановка задачі	25.02.2021-01.03.2021	виконано
3	Проведення дослідження	01.03.2021-01.04.2021	виконано
4	Підготовка пояснювальної записки	01.04.2021-01.05.2021	виконано
5	Підготовка презентації та доповіді	01.05.2021-08.05.2021	виконано
6	Перевірка на академічний плагіат	12 травня 2021р.	виконано
7	Нормоконтроль	12 травня 2021р.	виконано
8	Рецензування	17 травня 2021р.	виконано
9	Попередній захист	20 травня 2021р.	виконано
10	Занесення диплома в електронний архів	21 травня 2021р.	виконано
11	Допуск до захисту у зав. кафедри	24 травня 2021р.	виконано

Дата видачі завдання 25 січня 2021р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Власенко Л. А.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи : 54 с., 12 джерел, 10 рисунків.

АЛГОРИТМИ, АЛГОРИТМІЧНИЙ ТРЕЙДИНГ, ГЕНЕТИЧНІ АЛГОРИТМИ, ЕВОЛЮЦІЙНІ АЛГОРИТМИ, PYTHON

Об'єктами дослідження є генетичні алгоритми у сфері алгоритмічн торгівлі.

Метою роботи є використання генетичних алгоритмів для налаштування автоматичних торгових стратегій.

У результаті роботи був розроблений метод визначення найкращих налаштувань для визначення торгової стратегії шляхом застосування алгоритмів генетичного навчання. Розроблена програмна система, що реалізує запропонований метод.

ALGORITHMS, ALGORITHMIC TRADING, EVOLUTIONARY ALGORITHMS, GENETIC ALGORITHMS, PYTHON

Objectives of research are genetic algorithms in the field of algorithmic trading.

Aim of the work is to use genetic algorithms for automatic trading strategies.

As a result, a method was developed to determine the best settings for determining a trading strategy through the use of genetic learning algorithms. A software system that implements the proposed method has been developed.

Я, Чолак Костянтин Юрійович, студент гр. ІІЗм-19-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження генетичних

алгоритмів та їх застосування у сфері торгівлі», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

Вступ .....	8
1.1 Постановка задачі.....	10
1.1.1 Задача оптимізації .....	10
1.1.2 Методи оптимізації цільової функції.....	12
1.1.3 Метод еволюційних алгоритмів .....	14
1.1.4 Генетичні алгоритми.....	17
1.1.5 Проблеми генетичних алгоритмів .....	18
1.1.6 Торгова стратегія.....	19
1.2 Аналіз стану розв’язання задачі .....	24
2 Дослідження ефективності генетичних алгоритмів.....	26
2.1 План дослідження.....	26
2.2 Постановка мети дослідження .....	26
2.3 Визначення критеріїв ефективності рішення .....	27
2.4 Підготовка цінових рядів та вибір вхідних даних .....	28
2.5 Знаходження загального розв’язання для задачі .....	29
2.6 Знаходження часткового вирішення .....	30
2.7 Розглядання ручного налаштування .....	31
2.8 Застосування часткового розв’язання для знаходження загального .....	32
2.9 Аналіз результатів дослідження .....	33
2.10 Використання отриманих результатів.....	33
3 Вимоги про програмного забезпечення .....	35
3.1 Опис вимог до розробленої програми.....	35
3.2 Опис технологій та розробленої програми.....	36
Висновки .....	38
Перелік джерел посилань.....	39
Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії	Error! Bookmark not defined.

Додаток Б Звіт результатів Перевірки кваліфікаційної роботи на унікальність тексту .....	<b>Error! Bookmark not defined.</b>
Додаток В Програмний код .....	<b>Error! Bookmark not defined.</b>
Додаток Г Слайди презентації.....	<b>Error! Bookmark not defined.</b>
Додаток Д Наукові публікації .....	<b>Error! Bookmark not defined.</b>
Додаток Е Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимоги ДСТУ 3008: 2015 .....	<b>Error! Bookmark not defined.</b>

## ВСТУП

Природа завжди була великим джерелом натхнення для людства. Генетичні алгоритми — це алгоритми, засновані на природному відборі. ГА є підмножиною набагато більшої галузі обчислень, відомої як Еволюційне обчислення. Вони були розроблені Джоном Голландією та його студентами й колегами з Мічиганського університету, зокрема Девідом Е. Голдбергом, і відтоді з високим успіхом застосовувались для розв'язання різних проблем оптимізації.

Велика частина задач зводяться до задач оптимізації, часто задачі мають не одне, а багато рішень з різним рівнем оптимальності. Існує велика кількість алгоритмів для розв'язання задач оптимізації. Більшість з них базуються на переборі, зазвичай частковому. Еволюційні, зокрема генетичні алгоритми, якраз до таких відносяться.

Із появою технології автоматизованих систем розрахунку світ фінансів зробив великий крок уперед. Усе завдяки швидкості, яку надавали комп'ютери та складності логіки, яку можна задати алгоритмами. Алгоритми використовувалися для створення торгових програм, визначення цін на товари, як у сфері фінансів, так і сфері бізнесу [1].

Ламарк визначає еволюцію як виникнення нових форм шляхом перетворення старих. Саме так, поступово генетичні алгоритми вибирають найкращі форми, залишаючи невдалі варіанти.

Генетичні алгоритми неможливі без кодування поведінки форми у вигляді генів. Гени це одиниця спадковості, що відповідає за наявність або відсутність признаку. Набір генів одного організму або форми розв'язання проблеми називають генотипом. З впливом навколишнього середовища гени можуть мутувати — бути замінені на інші гени.

Як і у житті генетичні алгоритми отримують кращі результати при використанні принципу змішування генів — crossover. При створенні нового варіанту розв'язання задачі максимізації два варіанти змішують свої гени і новий

варіант отримає унікальний набір. Звичайно є шанс, що цей набір не буде життєздатним, проте при достатньому розміру популяції агенти мають великий шанс на вдалу комбінацію.

Метою даної роботи є теоретичний та практичний аналіз використання генетичних алгоритмів у сфері алгоритмічної торгівлі.

Наукова новизна представленої роботи полягає в розробці методів тонкого налаштування торгової стратегії за допомогою генетичних алгоритмів, використовуючи їх здатність до перенавчання, на відміну від чинних методів використання генетичних алгоритмів для прогнозування ціни або генерації торгових сигналів [2, 3, 4].

Публікації. Результати дослідження, проведеного в атестаційній роботі, були опубліковані в статті «Evolutionary algorithms for algorithmic trading» міжнародного наукового журналу «SWorldJournal».

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Постановка задачі

### 1.1.1 Задача оптимізації

Задача оптимізації базується на визначенні цільовій функції і пошуку максимального значення цієї функції. Цільовою функцією у даній роботі буде функція торгового алгоритму маркет-мейкінгу.

Функція приймає на вхід числові ряди із цінами фінансових інструментів у кожній точці часу (у роботі використовувалися хвилинні дані закриття) а також налаштування для стратегії, яка буде розглянута пізніше. Функція розраховує усі дії купівлі та продажу згідно з торговим алгоритмом та налаштуваннями, та на вихід віддає фінансовий результат.

Існують багато показників для опису торгової стратегії, або фінансового портфеля, такі як коефіцієнт Шарпа, бета-нейтральність, коефіцієнт Сортіно, коефіцієнт Трейнора, та інші. У цій роботі був використаний найпростіший — ітоговий фінансовий результат. Хоча він не відображає ризик, та максимізація цього показнику дуже наочна.

Задачею роботи є вивчення можливості застосування генетичних алгоритмів для максимізації описаної цільової функції (фінального фінансового результату).

Бувають ситуації коли одного показника мало, тоді застосовують багатоцільові функції, результати яких порівнюються за допомогою Парето-оптимальності [5].

Парето-оптимальності можна визначити як максимізацію якихось цілей при збереженні рівня оптимальності у інших цілей. Наприклад нам важливі кількість торгових угод, які провів робота та абсолютний дохід. У такому випадку еволюційним алгоритм оптимізуватиме один з цих показників лише за умови, що других не втратить. На рисунку 1.1 приведено приклад Парето-фронту —  $n$ -мірна матриця (де  $n$  дорівнює кількості показників), яка містить усі оптимальні рішення.

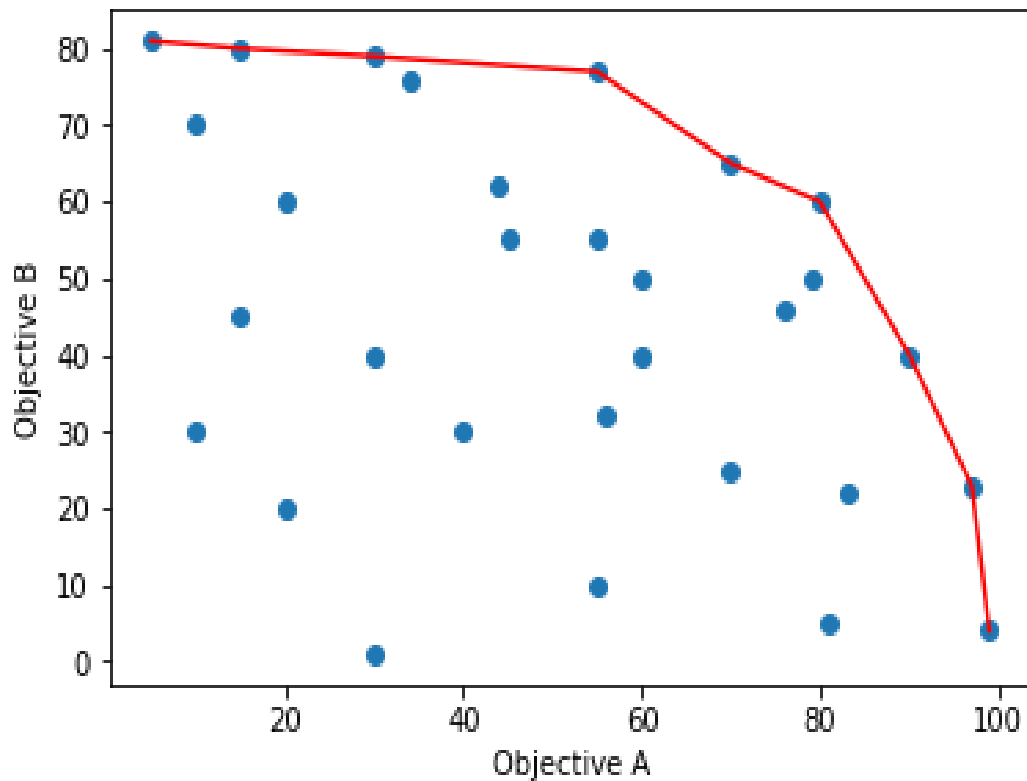


Рисунок 1.1 — Парето-фронт

Однією з проблем налаштування торгових стратегій є пов'язаність налаштувань між собою у часі та одна з одною. Налаштування алгоритму робляться не один раз, вони змінюються з часом, підлаштовуються під ринкову ситуацію. Тому налаштування можна сприймати як матрицю чисел, де кожний рядок це набір усіх налаштувань, а строки відображають плин часу. Аналітик може сказати, що зміна одного чи декількох налаштувань в одній точці часу до кращого результату, проте він не може сказати як би це вплинуло на усі наступні налаштування, чи було б оптимально змінити їх. А також не можна сказати, чи буде якась зміна налаштувань правильною, якщо аналітик після цього змінить попередні налаштування, оскільки вони пов'язані між собою.

Через проблематичність визначення оптимальних налаштувань важко використовувати стандартні методи генералізації правил, наприклад штучні нейронні мережі, бо вони потребують марковані дані тренування. Тому найкращим варіантом буде неконтрольоване навчання, наприклад еволюційний алгоритм.

### 1.1.2 Методи оптимізації цільової функції

Є велика кількість методів оптимізації цільової функції, кожен зі своїми сильними сторонами та недоліками, розглянемо деякі з них, виявимо проблеми застосування для розв'язання задачі максимізації обраного показника торгової стратегії та її налаштувань (ітоговий фінансовий результат).

Стохастичний градієнт спуску — ітераційний метод оптимізації значення цільової функції, використовує стохастичне наближення при пошуку. Цей метод використовує лише частину наявних тренувальних даних на кожній ітерації, які обирається випадково. Через це значення цільової функції будуть значно коливатися. Приклад стохастичного спуску зазначено на рисунку 1.2. Проте через стохастичну природу алгоритму він згодом зійдеться у локальному або глобальному мінімумі.

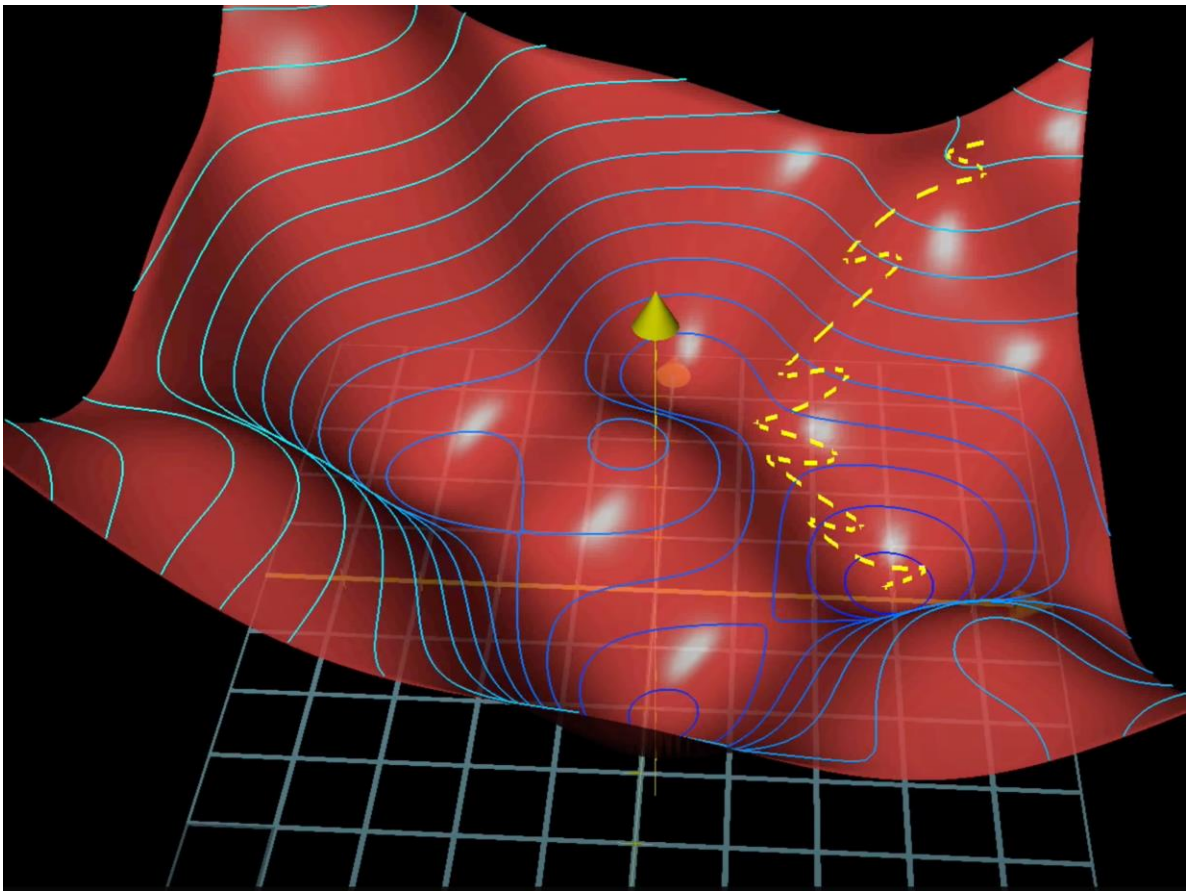


Рисунок 1.2 — Стохастичний спуск

Важливим фактором при пошуку мінімуму за допомогою стохастичного градієнтного спуску є фактор затухання швидкості навчання. Якщо з часом не зменшувати швидкість навчання, то градієнтний спуск буде постійно ходити навколо точки мінімуму.

Стохастичний градієнт спуску з моментумом — відрізняється від базового стохастичного градієнтного спуску тим, що гасить коливання та посилює направлений рух при пошуку мінімуму. Тим самим пошук прискорюється у правильному напрямку.

Адам — ітераційний метод, який використовує адаптивну оцінку моментуму. Цей метод увібрав у себе усе найкраще з попередніх методів оптимізації і вважається одним із найкращих методів оптимізації для загального використання на цей час. Проте йому як і іншим властиво застрягати на локальних мінімумах.

Симплекс-метод — метод для лінійної максимізації, який обходить повний перебір завдяки використанню проходу по вершинах геометричної фігури, яка образується, якщо нанести лінійні рівняння на площини розв'язків. Приклад алгоритму зазначено на рисунку 1.3.

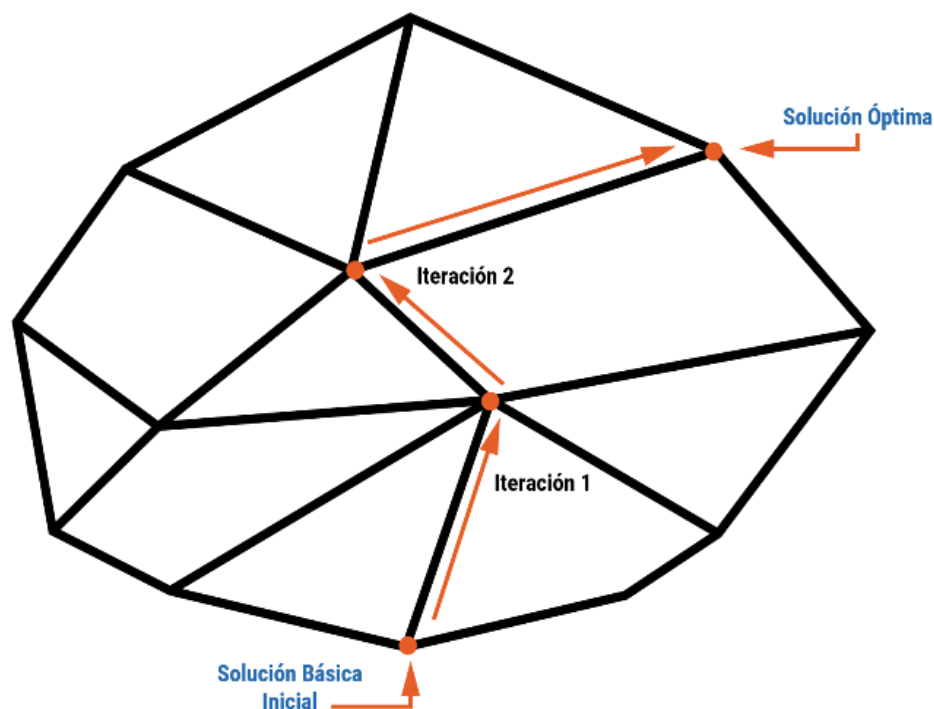


Рисунок 1.3 — Симплекс-метод

### 1.1.3 Метод еволюційних алгоритмів

Еволюційні алгоритми є підрозділом штучного інтелекту, який базується на імітуванні біологічної еволюції за теорією Чарльза Дарвіна для автоматичного знаходження рішень для різного роду задач.

У 1950-х роках принципи Чарльза Дарвіна стали застосовуватися при розв'язанні проблем. Протягом 1960-х та 1970-х років незалежні дослідження почали формувати те, що зараз відоме як еволюційні обчислення.

Як у біології, еволюційні алгоритми оперують особами видів (агентами), відбираючи тих, які показують себе найкращим чином (із максимальною цільовою функцією).

Процес навчання алгоритму генетичного відбору поділяють на наступні основні етапи:

- Створення початкової популяції;
- Відбір найкращих агентів;
- Мутування;
- Схрещення;
- Видалення ізольованих груп, які стагнують;
- Заміна видалених груп на нові.

Створення початкової популяції вимагає визначення структурного вигляду кожного агента. Усі агенти починають із випадковими генами.

Агенти розбиваються на ізольовані групи, це допоможе пришвидшити дослідження великої кількості локальних мінімумів. Через те, що групи ізольовані, то прогресія кожної групи схожа на прогресію усього алгоритму за відсутності груп. Таким чином групи дозволяють паралельно запустити алгоритм багато разів, автоматично перезапускаючи ті рази, які стагнують.

Наступним етапом є розрахунок значень цільової функції для кожного агента, на вхід цільова функція приймає цінові ряди разом із рядами налаштувань для кожної строки цін у ціновому ряді. Після застосування торгової стратегії з

указаними налаштуваннями на цінових рядах цільова функція повертає результат агенту.

Цільова функція викликається велику кількість разів (кількість поколінь помножена на кількість агентів), через це важливо оптимізувати її, бо найбільшу кількість часу алгоритм проведе саме на цьому етапі. Дана робота використовує мову Python, яка відома своєю низькою швидкістю, тому цільову функцію можна реалізувати на мові C, залежно від складності торгового алгоритму прискорення може бути на 2-3 порядки.

Відбір найкращих агентів відбувається після розрахунку значень цільової функції, та застосовується для виконання принципу “виживає найсильніший”, який тут визначається результатом цільової функції.

Вибирати найкращих агентів можна різними методами, найчастіше це робиться або прямолінійно, обираючи якийсь відсоток найкращих агентів, або випадково, з вірогідністю обрати агента пропорційною до значення його цільової функції у порівнянні з іншими агентами популяції, таким чином випадковість дає невеликий шанс не найкращим агентам показати себе ще раз, після мутування та схрещування.

Через те, що агенти розбиті на групи, то вибір проходить лише між агентами своєї групи. Таким чином агенти матимуть шанс на виживання у ситуації, коли вони початково показують не найкращий результат, але знаходяться поряд із більш оптимальним максимумом, ніж ті, що показуються кращі результати. Якщо ж усе таки група не була поряд із кращим локальним максимумом, то через декілька поколінь вона буде видалена через стагнацію. Це значно збільшує різноманітність агентів серед усіх груп. Без наявності груп усі агенти з часом стануть схожі на найкращого агента, який міг знайти локальний максимум. Цей спосіб допомагає перевірити якомога більше локальних максимумів, а не зупинитися на першому знайденому.

Мутування відбувається після відбору найкращих агентів. Із заданою вірогідністю агенти мутують, замінюючи заданий відсоток своїх генів на

випадкові. Цей етап дозволяє дослідити значно більшу область розв'язання задачі максимізації.

Проте не всі агенти піддаються мутаціям та схрещуванню, принцип елітизму пропонує не змінювати “елітних агентів”. Елітні агенти обираються як декілька найкращих агентів у популяції (або групі). Таким чином елітизм гарантує те, що найкращі гени популяції або групи не будуть втрачені через випадкову мутацію.

Для подальшого збільшення різноманітності агентів відбувається етап схрещування. Цей етап змішує попарно гени різних агентів із заданою вірогідністю, утворюючи нові види.

Коли ізольована група не показує нові результати (найкращий її агент) протягом декількох ітерацій, то доцільно видалити таку групу та замінити її на нову, створену випадковим чином. Завдяки такому підходу алгоритм може перебирати значно більше локальних мінімумів.

Процес ГА схематично представлений на рисунку 1.4.

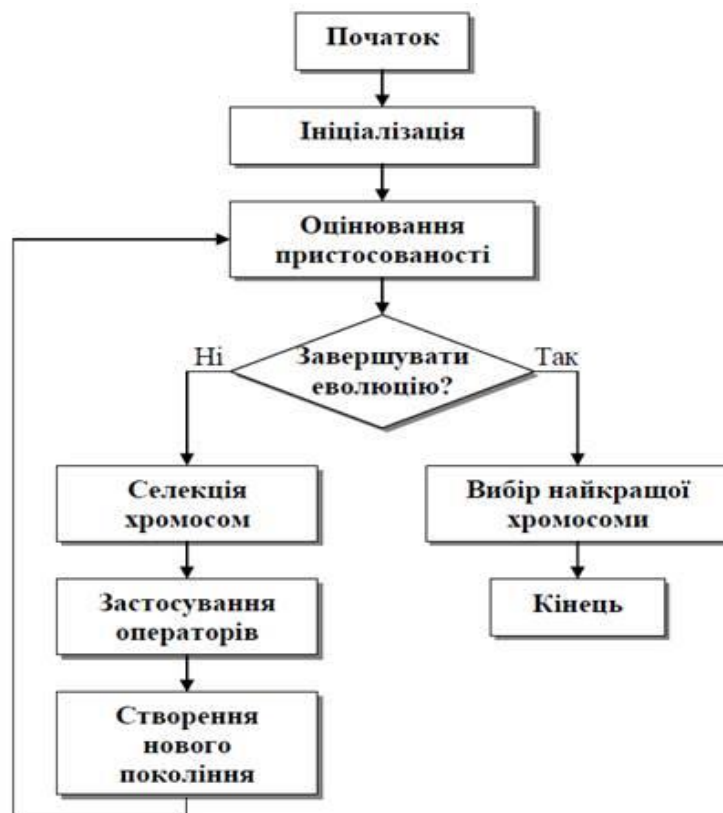


Рисунок 1.4 — Схема еволюційної оптимізації

### 1.1.4 Генетичні алгоритми

Генетичні алгоритми є підрозділом еволюційних алгоритмів. На відміну від базового алгоритму агенти задаються у вигляді набору генів [6]. Генотипом у генетичному програмуванні називають сукупність характеристик агенту, які описують його поведінку. Кожен агент має власний, часто унікальний генотип. Сукупність генотипів популяції утворює генофонд. Наявність різноманітності у генофонді значно впливає на швидкість знаходження рішень, через кількість локальних максимумів, які агенти перевірять.

Різнманітність генофонду схильна до зменшення через те, що невдалі комбінації генів через деякий час заміщуються вдалими, тим самим генофонд із часом складається лише із вдалих генотипів.

Існує декілька інструментів для впливу на різноманітність генофонду:

- США;
- Нормально-розподілена селекція;
- Ізолювання груп;
- Схрещування.

Нормально-розподілена селекція, окрім того, що дає шанс менш вдалим агентам проявити себе, ще й збільшує різноманітність генофонду, бо залишає не лише вдалі генотипи, як це робить порогова селекція, коли лише певний відсоток найкращих агентів відбираються для подальшого навчання.

Ізолювання груп ще менше дає можливість селекції знизити різноманітність, бо у кожній групі будуть свої найкращі агенти із різним ступенем пристосованості, і більш пристосовані агенти не витіснять менш пристосованих із часом.

Схрещування збільшує різноманітність генофонду завдяки змішуванню генотипів.

Усі ці методи працюють один із одним у симбіозі, бо нормально-розподілена селекція збільшує вірогідність того, що вдалий ген загально не

пристосованого агенту перейде до більше пристосованого, а не буде втрачений після селекції.

### 1.1.5 Проблеми генетичних алгоритмів

Еволюційні алгоритми автоматично не пропонують розв'язання для застрявання у локальних мінімумах, проте вони більш стохастичні і здатні перебрати більше варіантів за розумний час. Проте часто цього результату не вистачає через тонкість налаштувань алгоритмічної торгівлі.

Через це зазвичай цільова функція матиме велику кількість локальних мінімумів, різної оптимальності, через же важливо перебрати якомога більше з них, щоб визначити оптимальний. Тому в даній роботі запропонований метод, спрямований на збільшення різноманітності серед агентів у популяції, а через це й вірогідність зайти глобальний максимум.

Метод полягає в ізолюванні груп агентів один від одного, тим самим не буде одного агенту, який знайшов локальний мінімум. Натомість таких агентів буде по одному на групу, таким чином групи отримують можливість дослідити більше локальних мінімумів на предмет наявності глобального мінімуму.

Щоб пришвидшити пошук було запропоновано замінити новими групами ті, що не вдосконалюються протягом декількох ітерацій, таким чином алгоритм не обмежений початковою кількістю груп, він буде перебирати одну за одною у пошуках нових локальних мінімумів.

Також розглянемо проблему перенавчання, до якої схильні еволюційні алгоритми, тобто вони будуть давати правильні відповіді на тренувальних даних, проте при перевірці на нових даних вони даватимуть погані результати [7]. Перенавчання еволюційних алгоритмів є активно досліджуваною темою, єдиного розв'язання цієї проблеми досі не існує.

Залежність налаштувань одне від одного із плином часу створює велику

кількість локальних максимумів, через що багато із них будуть знаходити закономірності, які не працюватимуть для часових рядів, які генетичний алгоритм не бачив. Через це й виникає проблема перенавчання. У даній роботі розглядається метод розв'язання для цієї проблеми [8] — використання набору валідаційних цінових рядів.

У цій роботі перенавчання дало несподівані позитивні результати, замість того, щоб навчатися як налаштувати торгову стратегію, еволюційний алгоритм при перенавчанні буде шукати найкращі налаштування, які потрібно було встановити, проте не зможе узагальнити ці знання. Отримавши оптимальні налаштування можна використати їх як маркування даних для навчання штучної нейронної мережі [9].

#### 1.1.6 Торгова стратегія

Дана робота націлена на оптимізацію тонких налаштувань для активної торгової стратегії. Активну торгову стратегію відрізняє те, що вона націлена на те, щоб показати результат, кращий за ринковий (наприклад дати більший прибуток у порівнянні з індексом акцій S&P500, або краще співвідношення ризику до прибутку). Часто активні торгові стратегії є дельта-нейтральними, або приближеними до них. Це означає, що фінансовий результат не буде залежати від ринкової ситуації, і коли ринок впаде, то стратегія буде до цього готова, так не відчує це на прибутку.

Активна торгова стратегія також відрізняється великою кількістю торгових угод. Від сотень у день, до декількох тисяч у хвилину.

Торгові стратегії з тисячами угод у хвилину називають високочастотними (HFT — high frequency trading), таких стратегій велика кількість, кожна експлуатує недосконалість ринку, будь то арбітражна ситуація (коли один і той же товар на різних ринках, або різних секціях ринку має трохи різну ціну; при

достатньо низьких комісіях на цьому заробляють арбітражери), повільні торгові алгоритми, які можна прогнозувати, та забирати малу частку їх прибутку без ризику, та інші недосконалості.

Прикладами активних торгових стратегій є скальпинг, маркет-мейкинг, swing-trading, фрактальна торгова стратегія,

Стратегія скальпінгу розраховує на те, що програма встигне купити і продати актив по цінах пропозиції та попиту достатньо разів, щоб перевищити збиток від накопичення позиції, яка постійно повинна бути малою. Стратегія потребує дуже точного механізму фіксування збитків та низькі комісії, від яких залежить здатність алгоритму обернути якомога більший об'єм активу. Дана стратегія потребує низькочастотне обладнання та високошвидкісну торгову систему. Скальпер буде торгувати лише в одному напрямку — купівлі або продажу, залежно від внутрішньої логіки.

Стратегія скальпінгу активно використовується на валютах та облігаціях розвинених ринків, через їх відносну спокійність та великі об'єми ринку. Курси розвинених країн украй мало будуть коливатись у порівнянні із індексами акцій або товарами, і особливо окремими акціями.

Маркет мейкинг схожий на скальпинг, проте може створювати пропозицію як на покупку, так і на продажу, накопичувати значно більшу позицію (у тисячі разів більшу, ніж скальпер), та хеджувати її замість фіксування збитків. Хеджування це перекриття (зниження) ризику, прикладом хеджування є покупка опціону на актив, який треба захеджувати. У разі, якщо актив втратить у ціні (або навпаки виросте при наявності короткої позиції), то опціон компенсує збиток від основної позиції.

Swing trading базується на теорії трендів. Сигналом для входу в позицію є зміна тренду, яка призводить до підвищеної волатильності через те, що трендові гравці закривають свої позиції, нерідко примусово. Напрямок входу у позицію обирається з використанням фундаментального аналізу.

Фрактальна торгова стратегія використовує технічний аналіз для визначення точок входу та виходу із позиції. Технічним аналізом називають

використання статистичних інструментів для знаходження закономірностей у цінах та обсягах торгівлі або опису поточного стану ринку. Дана стратегія базується на припущенні, що певна послідовність японських свічок є сигналом на вхід у позицію, якщо ціна активу опуститься нижче центральної свічки. Японські свічки використовують для графічного відображення цінових рядів; через те, що у ринку за фіксований проміжок часу, скажемо хвилину, можуть відбутися сотні тисяч угод, то відображати кожну не є доцільним, японські свічки натомість відображають ціну першої, найдорожчої, найдешевшої та останньої угоди у встановлений час. Через це японські свічки часто називають OHLC свічками (open, high, low, close). Приклад японських свічок приведено на рисунку 1.5.

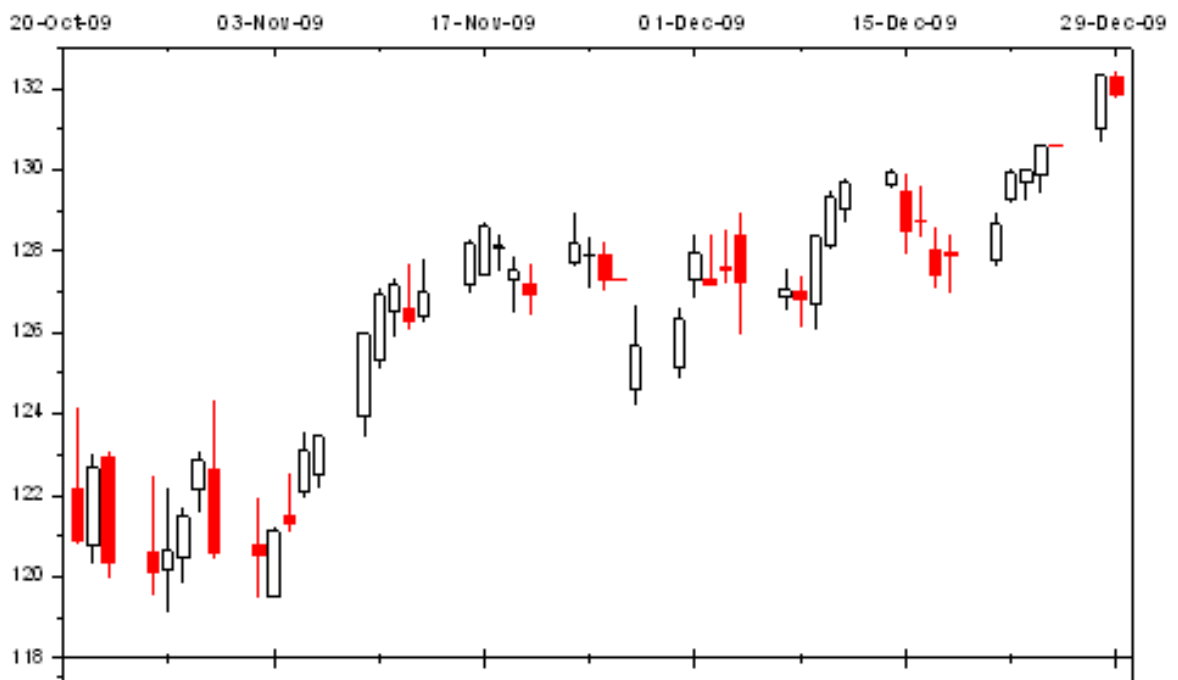


Рисунок 1.5 — Приклад японських свічок.

Червоними відображені свічки, за час якої ціна знизилася, тобто ціна відкриття більша за ціну закриття. Білим відображені свічки росту ціни. Їх ще часто позначають зеленим кольором.

Кожна свічка складається із тіла (заповнений або пустий прямокутник) та хвоста (вертикальна лінія, що проходить через тіло). Тіло свічки відображає першу та останню ціну. Якщо свічка червона (тобто ціна падає), то зверху буде

перша ціна, а знизу буде остання ціна. Хвіст завжди показує максимальну ціну зверху та мінімальну знизу. Хвіст може сховатися за тілом свічки, якщо найнижча та найвища ціна проміжку часу не виходила за ціни відкриття та закриття.

Перед тим, як розглянути торгову стратегію, яка була взята як об'єкт дослідження на факт можливості використання генетичних алгоритмів для максимізації встановленого показника треба описати базові поняття у торгівлі, такі як стакан, спред у стакані, маркет мейкер, маркет тейкер.

Стакан — набір заявок на покупку та продаж. Якщо ціни заявок на покупку та продажу перетинаються, то відбувається угода. Якщо ціна заявки не перетинається з іншими, то вона буде у стакані доки не перетнеться або не буде скасована. Приклад торгового стакану відображений на рисунку 1.6.

153 320	36
153 310	29
153 300	63
153 290	49
153 280	126
153 270	34
153 260	37
153 250	56
153 240	8
153 230	13
153 220	25
153 210	33
153 200	47
153 190	30
153 180	27
153 170	35
153 160	67

Рисунок 1.6 — Приклад торгового стакану

Спред у стакані — різниця між найнижчою ціною продажу та найвищою ціною покупки.

Маркет мейкер — сторона угоди, яка створила пропозицію, тобто перша подала торгову заявку у стакан.

Маркет тейкер — сторона угоди, яка створила попит, тобто другою подала торгову заявку у стакан.

У даній роботі буде розглянута рівнева стратегія. Суть стратегії полягає у тому, що збільшувати позицію коли ринок йде проти нас, та зменшувати у протилежній ситуації, заробляючи на кожному зменшенні позиції. Стратегія базується на припущенні, що спред у стакані та активність на ринку буде більша, ніж абсолютний рух ціни. Параметрами такої стратегії можуть бути розмір рівню (обернена частота набору позиції), максимальна позиція, параметри фіксування збитків, об'єми заявок та інші. Стратегія може працювати на покупку активу, або на продаж активу. При зменшенні частоти набору позиції стратегія працюватиме менш агресивно, через що знизиться прибуток, проте ризик також буде менший. Разом із параметрами фіксування збитку з'являється можливість балансувати між ризиком та прибутком, або брати більший ризик із можливістю повністю закрити позицію та прибрати ризик.

Більшість стратегій не будуть приносити прибуток без тонкого налаштування. Через це область рішень для визначеної задачі матиме велику кількість локальних максимумів та важливо буде обійти якомога більше з них.

Дана робота не бере до уваги такі фактори як торгова та брокерська комісія, виконання угоди по поганій ціні через затримки між даними, які бачить стратегія і реальними цінами, а також через затримку у посилянні торгових заявок.

## 1.2 Аналіз стану розв'язання задачі

Поставлена задача не має аналогів вирішень. Натомість є велика кількість прикладів використання еволюційних/генетичних алгоритмів для прогнозування цін та створення торгових стратегій [2, 3, 4]. Дана робота націлена на налаштування все розробленої стратегії. Через те, що відсутність фіксованої стратегії значно знижує шанси на можливість формалізування її логіки. А через те, що генетичні алгоритми часто схильні до перенавчання, то замість формалізування логіки ми отримуємо набір торгових рішень (купівля, продаж), які будуть оптимальними для того проміжку часу, на якому навчався алгоритм, формалізувати ці рішення у стратегії у край важко. Також запропоноване рішення поставленої задачі не намагається знайти спосіб генералізувати налаштування стратегії. Натомість алгоритм видасть найкращі налаштування для даної аналітиком/трейдером стратегії, що значно простіше генералізувати, ніж дії купівлі та продажу.

Розглянемо роботу “Evolutionary Algorithm in Forex trade strategy generation”, вона пропонує використання еволюційних алгоритмів для відкриття довгих та коротких (покупки та продажу) маржинальних позицій виходячи з рішень генетичного алгоритму, що базуються на технічних індикаторах. На відміну від даної роботи, стаття “Evolutionary Algorithm in Forex trade strategy generation” розглядає еволюційний алгоритм як саму стратегію. Разом із тим, що робота навчала алгоритм на маленькому обсягу даних (два тижні 10-ти хвилинних цін), то не може бути впевненості у тому, що алгоритм так само матиме можливість генералізувати показники технічного аналізу для генерації рішень покупки та продажу.

Також існують підходи, у яких стратегія налаштовується на етапі створення, і під час торгівлі налаштування не змінюються. Такий підхід підійде для простих стратегій, як описано у статті “How to Optimize Your Strategy Parameters With Genetic Algorithms?” [10]. Основна різниця із запропонованим

методом у даній роботі полягає у тому, що запропонований метод змінює налаштування у реальному часі, дозволяючи підлаштовуватися до риночної ситуації.

Дана робота є унікальною через те, що вона не пропонує спосіб генералізації торгових правил виходячи з торгової ситуації, а знаходить найкращі налаштування у конкретному проміжку часу, які потім можна використовувати у вже чинних рішеннях для генералізації цих торгових правил, таких як штучні нейронні мережі (recurrent neural networks, dense neural networks, та інші). Наявні роботи натомість пропонують використання генетичних/еволюційних алгоритмів для передбачення ціни або генерації торгових сигналів, що не дає фундаментальної впевненості у стратегії, до якої прийде генетичний алгоритм. Коли торгова стратегія фіксована і у генетичного алгоритму є вплив лише не її налаштування, то можна фундаментально підтвердити працездатність стратегії та мати впевненість, що вона не перестане працювати при найменших змінах у торговій ситуації.

## 2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ГЕНЕТИЧНИХ АЛГОРИТМІВ

### 2.1 План дослідження

Дослідження складається з наступних кроків:

- Постановка мети дослідження;
- Визначення критеріїв ефективності рішення;
- Підготовка цінових рядів та вибір вхідних даних;
- Знаходження загального розв'язання задачі;
- Знаходження конкретного рішення;
- Розглядання ручного налаштування торгової стратегії;
- Порівняння ефективності рішень.

### 2.2 Постановка мети дослідження

Метою даного дослідження є виявлення ефективності застосування генетичних алгоритмів для розв'язання задачі ручного налаштування торгових алгоритмів. Дане дослідження розгляне три види налаштування:

- генетичним алгоритмом;
- генетичним алгоритмом із вбудованим багат шаровим перцептроном;
- ручне налаштування.

Після розробки та тестування кожного із методів буде порівняно ефективність за визначеними критеріями. Дослідження перевірить наскільки можливо генетичним алгоритмом із вбудованим багат шаровим перцептроном отримати загальне рішення, тобто рішення, яке буде давати високий результат як для даних, на яких воно було натреновано, так і на нових даних.

Результатом дослідження буде визначення ефективності використання генетичних алгоритмів у порівнянні з експертним налаштуванням.

Налаштуваннями обраної стратегії будуть — частота набору позиції (при збільшенні негативно впливає на ризик, проте позитивно впливає на зафіксований прибуток), та фіксування збитку (закриття поточної позиції, негативно впливає на зафіксований прибуток, вбирає ризик). Складність налаштування полягає у балансуванні ризику та прибутку та пристосуванні до ринкової ситуації.

### 2.3 Визначення критеріїв ефективності рішення

Критеріями роботи торгового алгоритму (а отже і його налаштування) є фінансовий результат та загальність рішення. Через те, що на ефективність стратегії сильно впливає ринкова ситуація, то був узятий період, який включає проміжки часу із високою та низькою волатильністю. Але найбільший вплив має тренд ціни за обраний період, для виключення його впливу на результати стратегія перевіряється окремо на покупку нафти, та на продаж. Це дозволило оцінити результати стратегії як суму двох направлених стратегій та виключити ціновий тренд. Якщо стратегія дуже гарно покаже себе на періоді росту, то вона так само погано покаже себе на періоді падіння за виключенням зафіксованого прибутку, який й показуватиме ефективність налаштування.

Через те, що нафта торгується на різних біржах, і на кожній множник вартості свій, то замість доларового фінансового результату у даній роботі буде порівняно фінансовий результат у пунктах, які потім можна перевести у долари за множником обраної біржі (1000 на CME, 10 на MOEX і т.д.).

## 2.4 Підготовка цінових рядів та вибір вхідних даних

Всі дослідження були проведені на цінових рядах на нафту марки Brent, які були взяті з відкритого джерела `finam` [11]. Нафта торгується лише у вигляді ф'ючерсів (договору на майбутню поставку товару за фіксованою ціною), у роботі були використані цінові ряди на грудневий ф'ючерс 2020 року, бо у цей період ціна на нафту коригувалася після активного росту у післякризовий період, через що обрані цінові ряди мають як падіння, так і ріст ціни (див. рис. 2.1).

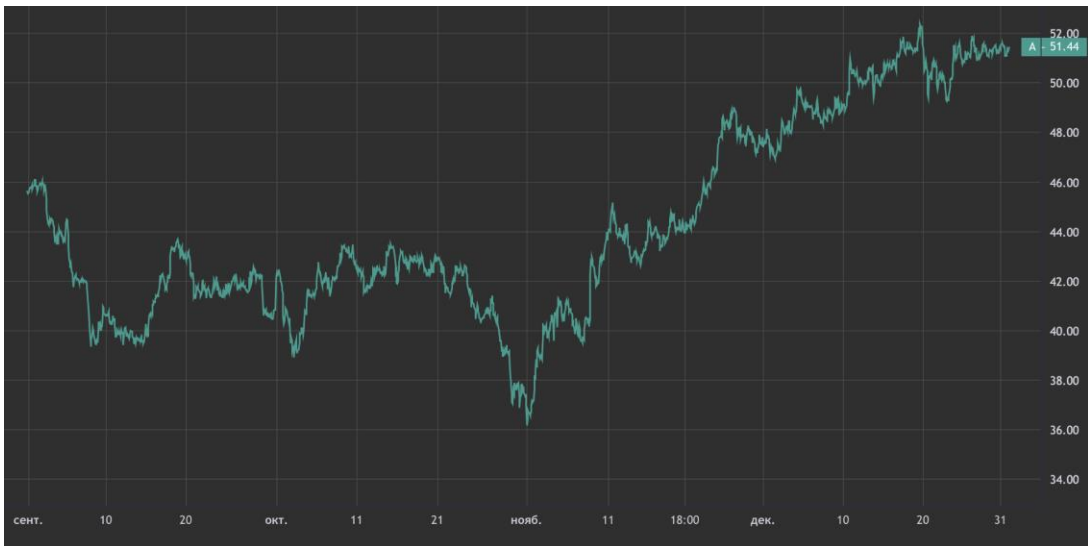


Рисунок 2.1 — Обраний ціновий ряд

Для пошуку загального рішення потрібні преобразовані дані, так як самі по собі точки цінового ряду не можуть нести корисної інформації. Найпростішою індикатор, який додасть скільки-небудь корисну інформацію є різниця між точками цінового ряду  $P$  (Формула 1.1) .

$$K_i = A_i - A_{i-1}$$

Формула 1.1 — Індикатор зміни ціни

Існує велика кількість більш інформативних та складних індикаторів, у даній роботі були використані Simple Moving Average (SMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Exponential Moving Average (EMA). Ці індикатори потрібні для знаходження загального рішення. Загальне рішення являє собою логіку налаштувань виходячи із значень індикаторів.

Simple Moving Average, через свою простоту, є найбільш популярним з них (формула 1.2), цей індикатор зазвичай попередньо встановлений на більшість графіків цін. Він розраховується як середнє значення  $n$  точок цінового ряду, часто використовують декілька SMA (із різними  $n$ ) одночасно, це дозволяє побачити картину у різних масштабах.

$$K_i = \frac{1}{n} \sum_{j=1}^n A_{i-j}$$

Формула 1.2 — Індикатор Simple Moving Average

Relative Strength Index є індикатором моментуму, він оцінює силу руху ціни для визначення завищення або заниження ціни. Значення індексу коливаються від нуля до ста.

Moving Average Convergence Divergence — індикатор, що показує різницю між двома експоненційно середніми цінами активу. Як і RSI є індикатором моментуму, проте дозволяє оцінити силу тренду, а не перекупленість ціни, як це робить RSI.

Exponential Moving Average дуже схожий на Simple Moving Average, відрізняється тим, що новіші ціни мають більшу вагу, ніж старі.

## 2.5 Знаходження загального розв'язання для задачі

Загальне розв'язання відрізняється від часткового можливістю застосування на невідомих для алгоритму часових рядах. Загальність рішення забезпечується визначенням логіки між ринковою ситуацією та оптимальними налаштуваннями для торгового алгоритму. Ввівши перцептрон між вхідними даними (ринкову ситуацією) та вихідними даними (налаштуваннями) агенти можуть впливати на поведінку стратегії лише запроваджуючи правила. Результати загального розв'язання зображені на рисунку 1.7. Синя лінія показує результат найкращого агента. Помаранчева показує середній результат вибірки.

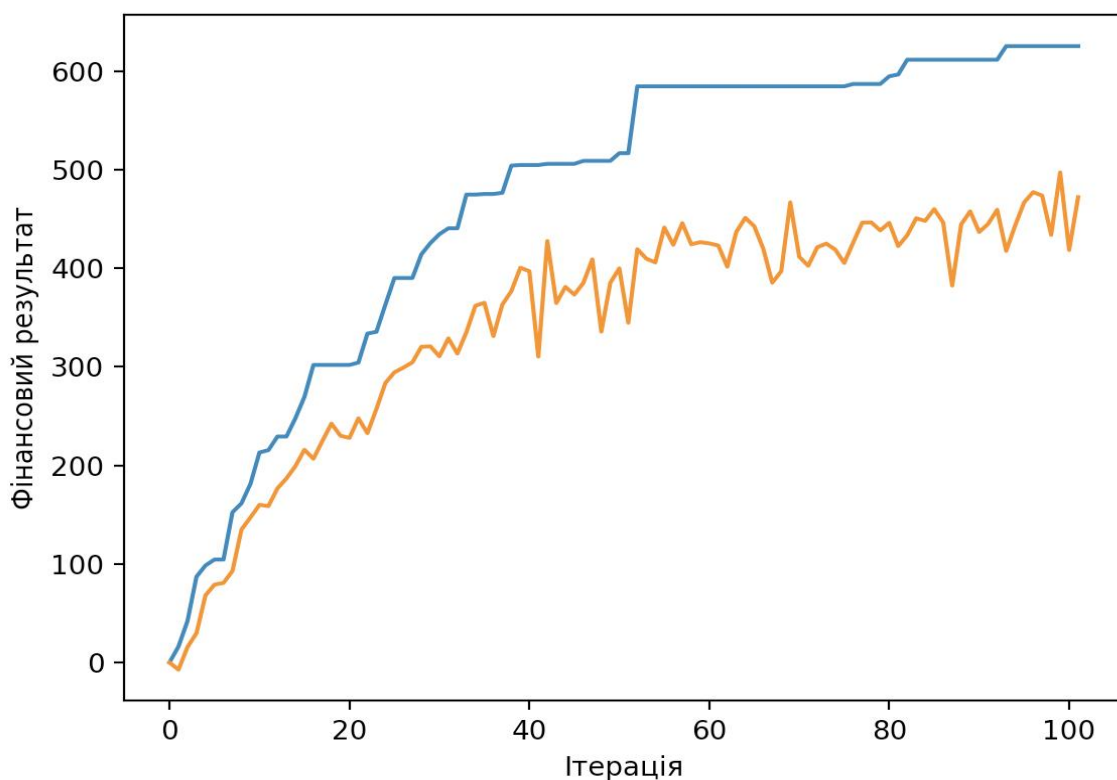


Рисунок 2.2 — Результати генетичного алгоритму із перцептроном

Перевірка на валідаційних даних показала зменшення результатів на 34 відсотки, з 625 до 413. Це вказує на те, що алгоритм не повністю, проте вивів правила, що знаходять загальне рішення для поставленої задачі.

## 2.6 Знаходження часткового вирішення

Якщо дати агентам можливість напряму впливати на налаштування, без необхідності виводити правила, тоді агенти зможуть знайти більш оптимальні налаштування. Ціною за оптимальність буде відсутність загального рішення налаштування, алгоритм не надасть відповіді на питання “як оптимально налаштувати торгову стратегію”, а лише “які налаштування були оптимальними для заданої ринкової ситуації”. Проте, завдяки подальшій обробці з цих даних можна отримати загальне рішення. Гени у цьому варіанті напряму транслюватимуться у налаштування і для кожної точки часу будуть свої гени. Це і дозволить знайти оптимальні налаштування. Результати генетичного алгоритму без перцептронів вказані на рисунку 2.3.

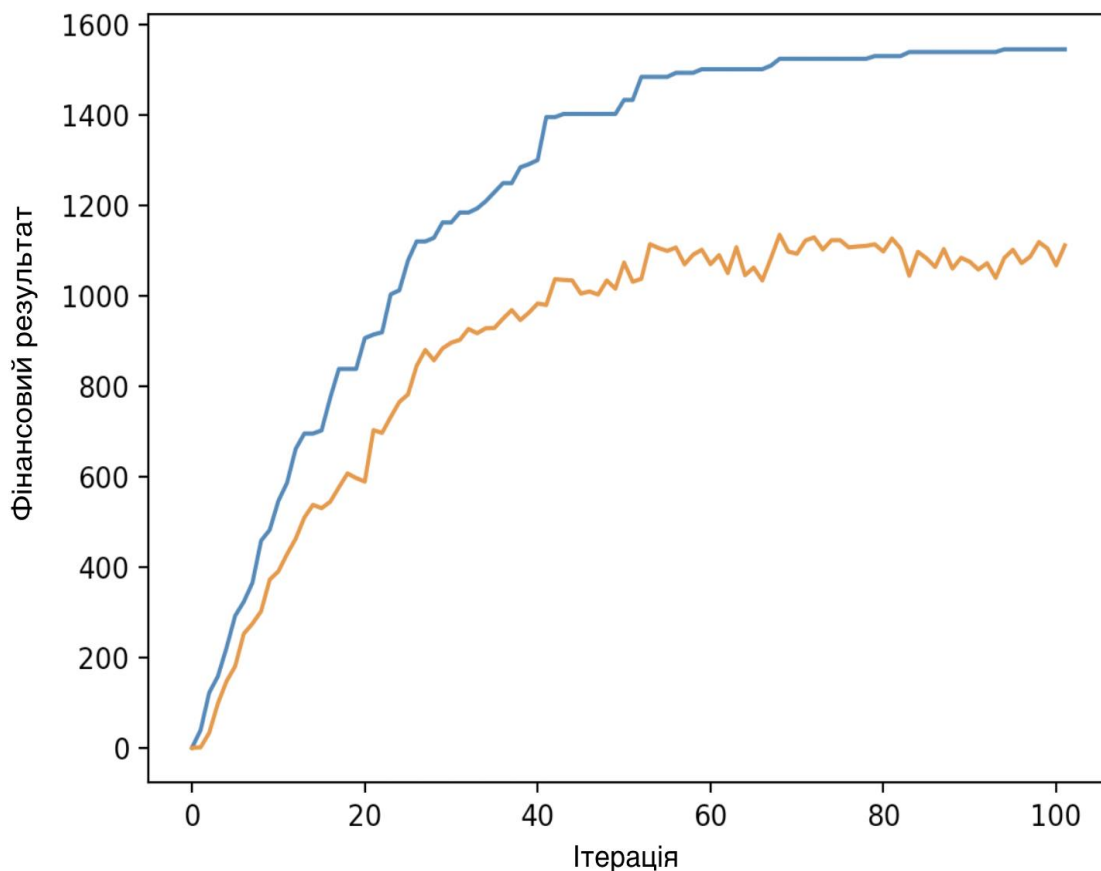


Рисунок 2.3 — Результати генетичного алгоритму

## 2.7 Розглядання ручного налаштування

Для порівняння автоматичного та ручного налаштування стратегія була налаштована вручну, та перевірена на тому самому проміжку часу. У ході експерименту стратегія була налаштована за наступним принципом — коли ціна робить сильні рухи (більше за  $N\%$ ), то параметр частоти набору треба зменшити на  $M$  пунктів, щоб кратно не підвищувати ризик. При русі більшому за  $K\%$  позиція повністю закривалась (фіксування збитку). Найкраще себе показали такі набори параметрів  $M=0.01$ ,  $N=1.3\%$ ,  $K=3\%$ . При таких налаштуваннях фінансовий результат складає 273 пункти. Цей результат не є позитивним, через те, що тестування відбувається в ідеальному середовищі — без комісій, затримок мережі та торгового алгоритму. При виході з ідеального середовища результат може стати зі знаком мінус.

## 2.8 Застосування часткового розв'язання для знаходження загального

Отримавши оптимальні результати налаштувань за допомогою метода знаходження часткового рішення можна використати як марковані дані для навчання за допомогою штучної нейронної мережі. Різниця між загальним рішенням полягає у тому, що перцептрон не буде вставлений у генетичну модель, натомість перцептрон тренується за допомогою алгоритму зворотного поширення. Ціль цього алгоритму зменшити помилку. У цій роботі була використана помилка Mean Squared Error (Формула 2.1). Де  $N$  це кількість вагів на останньому шарі (кількість виходів мережі),  $y_i$  це очікуєий результат,  $\hat{y}_i$  — отриманий результат.

$$E = \frac{1}{2N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

## Формула 2.1 — Помилка MSE штучної нейронної мережі

Зменшення помилки досягається за допомогою визначення градієнту кожного шару перцептрону та виправлення вагових і порогових коефіцієнтів на градієнт помножений на помилку. Помилка першого шару була розрахована за формулою 1.1, для кожного попереднього шару (алгоритм рухається з кінця) за помилку було взято ваги наступного шару помножені на його градієнт.

Результат генералізації залежить від етапу, на якому він був проведений. Якщо провести його занадто пізно, коли агенти знайшли багато унікальних особливостей навчального проміжку часу, то генералізація дасть гірші результати.

Після генералізації та перевірки на тестових даних було отримано фінансовий результат 651 пункт.

### 2.9 Аналіз результатів дослідження

Найкращі результати за обраними критеріями показав метод генетичного алгоритму без перцептрону із подальшим виділенням правил для генералізації налаштувань. Виявилось важливим зупинити навчання до виходу у плато, де генетичний алгоритм дуже гарно вивчить тренувальні ринкові дані, роблячи процес генералізації менш ефективним. Генетичний алгоритм із перцептроном показав гірші результати за критерієм абсолютного фінансового результату через обмеженість у налаштуваннях згідно до знайдених правил, що не давало знайти найбільш оптимальні комбінації налаштувань. Результат ручного налаштування у порівнянні з автоматизованими налаштуваннями показує себе значно гірше.

### 2.10 Використання отриманих результатів

У ході дослідження було отримано алгоритм для знаходження часткового розв'язання задачі максимізації для фінансового результату торгового алгоритму за допомогою тонких налаштувань. Хоча запропонований алгоритм напряду не дає загального рішення, проте отриманий результат можна використати для його отримання через примінення штучної нейронної мережі, як було показано у демонстраційній програмі.

Штучні нейронні мережі потребують наявності тестових даних (ряди цін) та “правильні відповіді” — налаштування, що дають максимальний фінансовий результат.

Проте знайти оптимальні налаштування для спеціаліста не проста задача, бо налаштування у часі  $T_n$  впливатимуть на всі наступні налаштування ( $T_2...$ ), А усі попередні налаштування ( $T_0...T_{n-1}$ ) впливають на  $T_n$ . через це зміна налаштувань в одній точці часу потребує перерахунку у всіх наступних точках часу.

Запропонований метод дозволяє знайти ці самі оптимальні налаштування без кропіткої роботи експерта у даній сфері. Це дозволяє використання штучних нейронних мереж для знаходження загального рішення, яке еволюційним методом знайти маловірогідно через велику схильність алгоритму до перенавчання, яка й була використана у цій роботі.

## 3 ВИМОГИ ПРО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Опис вимог до розробленої програми

Перед початком роботи над програмною системою необхідно визначитися з вимогами до неї. У загальному випадку під вимогами розуміють сукупність властивостей, які повинна мати система, що реалізується.

Розроблювана система призначена для використання професійним алгоритмічним трейдером або аналітиком. Система повинна надавати такий функціонал:

- надати користувачу можливість задавати цільову функцію, яка визначить якість рішень агенту;
- надати користувачу можливість задати всі ознаки генотипу, щоб він відповідав налаштуванням торгової системи;
- симулювати торгові заявки та угоди;
- уміти порівнювати агентів та відбирати найкращих з них;
- надавати користувачу можливість задавати багатоцільові задачі;
- інтегруватися з різними джерелами ринкових даних.

Алгоритм генетичного навчання повинен бути реалізована як бібліотека, таким чином його можна буде інтегрувати у вже реалізовані аналітичні системи, або без зайвих зусиль запустити з потрібними змінами, бо даний алгоритм сильно гнучкий у налаштуваннях, тому й реалізація повинна бути такою ж.

Через те, що реалізація алгоритму представлена у вигляді бібліотеки, то вимоги до пристрою, де представлена бібліотека буде використовуватися доволі низькі та обумовлені використанням програмної мови Python:

- ОС Windows \* / MacOS \* / Linux \*;
- ОЗУ 4 Гб або більше;
- 1 Гб вільного простору на жорсткому диску.

Також, через реалізацію у вигляді бібліотеки, система не потребує наявності БД для зберігання ринкових даних, це залишено на розсуд користувача.

Призначення розробки. Реалізація призначена для оптимізації налаштування стратегій для торгівлі на фондових, срочних, валютних, криптовалютних ринках. Результатом проведеної роботи стане гнучкий інструмент для проведення генетичного навчання.

Особливості системи. Основними функціями які містить програмний продукт є пошук оптимальних налаштувань стратегії у заданій користувачем ринковій ситуації. Стратегія задається через створення цільової функції. На виході бібліотека віддаватиме результат торгової стратегії та оптимальні налаштування для кожної точки часу.

### 3.2 Опис технологій та розробленої програми

У ході роботи була реалізована програмна бібліотека для оптимізації налаштувань торгових стратегій. Програма написана на програмній мові Python, та використовує одну сторонню бібліотеку `numru`, яка реалізує векторні математичні операції на програмній мові C для прискорення виконання обчислень. Завдяки принципу Single Instruction Multiple Data (SIMD) на процесорному рівні бібліотека `numru` прискорює векторні/матричні обчислення у сотні разів. Бібліотека надає абстрактний клас оптимізатора, від якого можна наслідуватися, реалізувати методи, які описують структуру агента, цільову функцію, дозволяють змінити кожен шаг еволюційного алгоритму, такі як відбір найкращих агентів, мутування, елітизм, схрещування, ізольоване групування та інші. Був використаний підхід багатосарової складності. Можна використовувати бібліотеку не розбираючись у деталях реалізації, та навіть у деталях еволюційного/генетичного алгоритмів, проте при необхідності можна налаштувати та змінити майже усе.

Еволюційний алгоритм у ході навчання викликає цільову функцію багато разів (кількість поколінь помножена на кількість агентів), через що важливо оптимізувати усі операції, які у ній відбуваються. Для оптимізації потрібно розібратися із тих, який тип операцій буде відбуватися й займатиме більшу частину процесорного часу. Через те, що еволюційний алгоритм використовується для оптимізації торгової стратегії, то він буде оброблювати велику кількість цінкових рядів, які можна представити у вигляді вектора. Найпопулярнішим методом оптимізації векторних розрахунків на програмній мові Python є бібліотека `numpy`, яка прискорює виконання програми одночасно (паралельно) обробляючи дані завдяки принципу SIMD [12], який описаний на рисунку 3.1. Одна процесорна команда одночасно робить однакову операцію над декількома словами даних. Через описані причини для пришвидшення виконання алгоритму була використана бібліотека `numpy`.

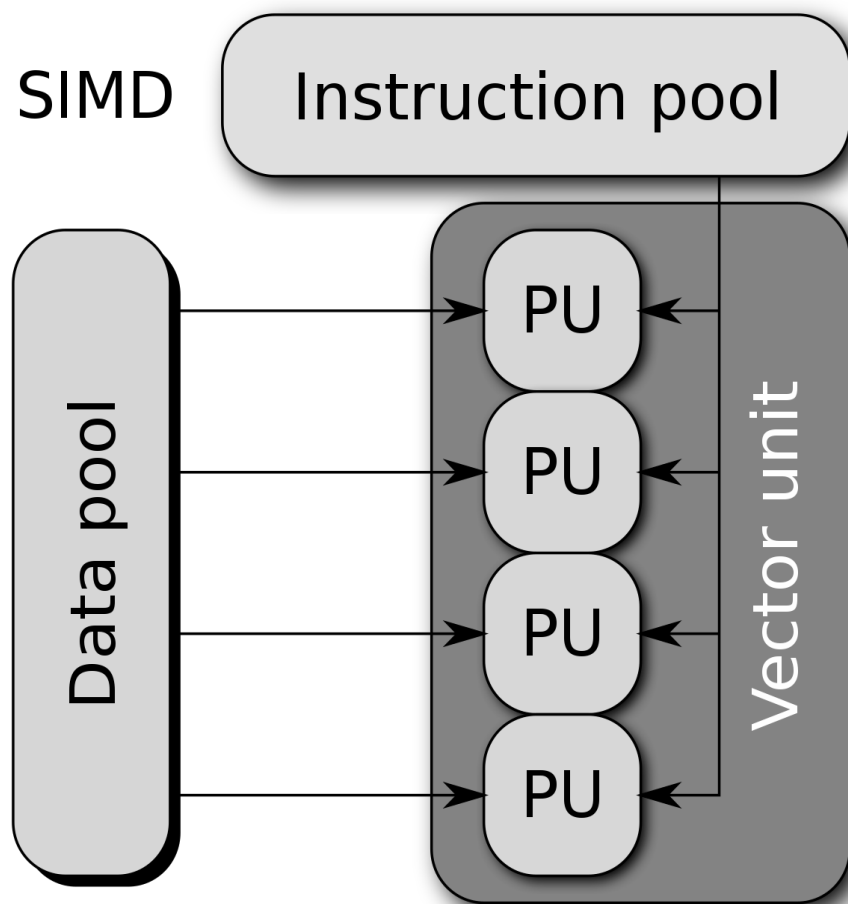


Рисунок 3.1 — Single Instruction Multiple Data

## ВИСНОВКИ

У результаті виконання роботи було спроектовано та реалізовано програмне забезпечення у вигляді підключаемого пакету для оптимізації налаштування торгових стратегій за допомогою генетичного алгоритму для використання на фондових, срочних, валютних, криптовалютних ринках.

У ході роботи було проведено аналіз предметної області, пов'язаної з використанням генетичного алгоритму, а також було розглянуті проблеми тонкого налаштування торгових стратегій. Виконано пошук та порівняння оптимізаційних алгоритмів. Покладаючись на отримані висновки, сформовано базовий набір вимог до програмного забезпечення.

У ході практичної частини роботи було сформовано критерії оцінки ефективності налаштування торгових алгоритмів та порівняно три способи налаштування: ручний, за допомогою генетичного алгоритму, та за допомогою генетичного алгоритму із перцептроном.

За результатами проведених досліджень можна зробити висновок, що генетичні алгоритми дозволяють автоматизувати процес налаштування торгових стратегій і роблять це краще за людей.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Forecasting of airfare prices using time series Gordiievych, A. , Shubin, I. 2015 Information Technologies in Innovation Business Conference, ITIB 2015 - Proceedings, 2015, с. 68-71, 7355055.
2. Using Genetic Algorithms to Build Trading Strategies - URL: <https://towardsdatascience.com/using-genetic-algorithms-to-build-stock-trading-strategies-d227951d3df0> (дата зверення 12.02.2021).
3. An Evolutionary Method for Financial Forecasting in Microscopic High-Speed Trading Environment - URL: <https://www.hindawi.com/journals/cin/2017/9580815/> (дата зверення 12.02.2021).
4. Evolutionary Algorithm in Forex trade strategy generation URL: <https://annals-csis.org/proceedings/2010/pliks/143.pdf> (дата зверення 12.02.2021).
5. Garcia S. Comparison of Multi-Objective Evolutionary Algorithms to Solve the Modular Cell Design Problem for Novel Biocatalysis / Sergio Garcia // 2019. – 13 с.
6. Alain Pétrowski Evolutionary Algorithms / Alain Pétrowski // 2017. – 236 с.
7. Exploring Overfitting in Genetic Programming URL: [https://link.springer.com/chapter/10.1007/978-3-540-24621-3\\_22](https://link.springer.com/chapter/10.1007/978-3-540-24621-3_22) (дата звернення 12.02.2021).
8. Нейронні мережі і генетичні алгоритми / Кононюк А. Ю. // м. Київ 2008. – 470 с.
9. The Handbook of Brain Theory and Neural Networks: Second Edition / Michael A. Arbib / A Bradford Book // 2002. – 1344 с.

10. How to Optimize Your Strategy Parameters With Genetic Algorithms?  
URL: <https://medium.com/swlh/how-to-optimize-your-strategy-parameters-with-genetic-algorithms-23671d7b446c> (дата звернення 12.02.2021).
11. Фінам - URL: <https://www.finam.ru/> (дата звернення 12.02.2021).
12. NumPy SIMD Optimizations - URL:  
<https://numpy.org/devdocs/reference/simd/simd-optimizations.html> (дата звернення 12.02.2021).